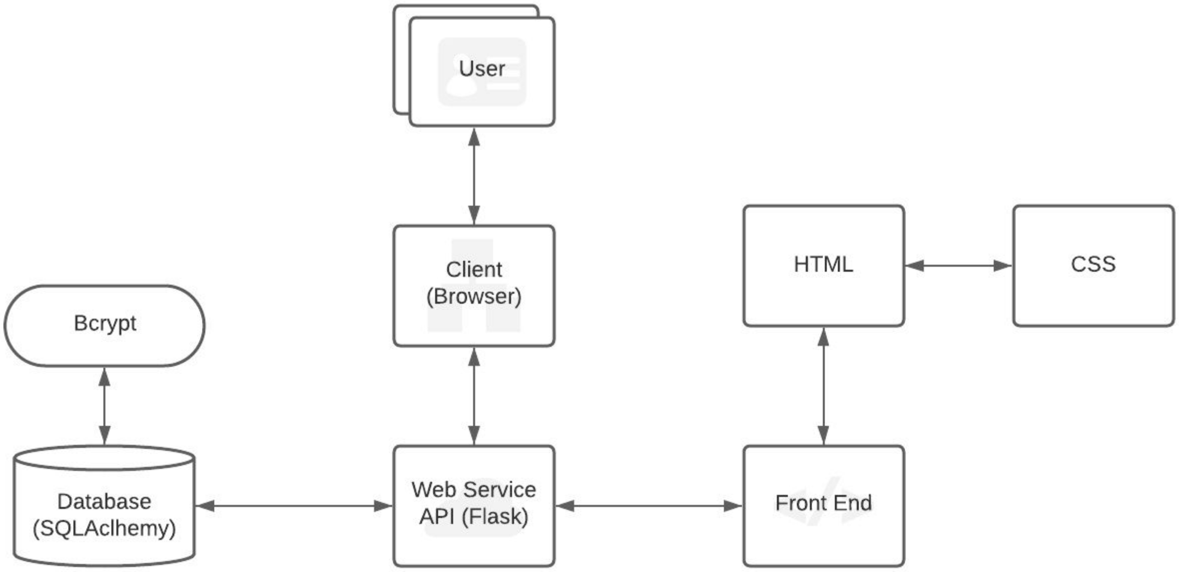
Team 203-4: Dumpster Devs
Zoe Stewart
David Dayan
John Lee
Makayla Johnson
Quinn Stone
Mitchell LaRocque

Revised List of Features

Priority of Development	Status	Feature	Description
1	**Completed**	**Profile Creation**	Ability for users to create a unique and personal profile for themselves in order to save their list of favorite players/data, and/or save their own personal stats/comparisons for their sport.
2	**Completed**	**Download Data**	Download data as a specified file type.
3	**In progress**	**Display Data**	Pulling stats and presenting them within the site under various categories (by sport, by better/worse, etc)
4	**In progress**	**Compare and Visualize Data (Graphs)**	Ability for users to select a few players of the same sport in order to compare and visualize stats in graph or other format. Also display graphs for team based metrics.
5	**Modified scope/in progress**	**Share Data**	Assign unique ID and link to each graph depending on parameters, share to others. Also allow for export to social media apps.
6	**Modified sope: not started**	**Sports Rankings**	Basic ranking based on multiple stat categories instead of complicated sports analytics algorithm.

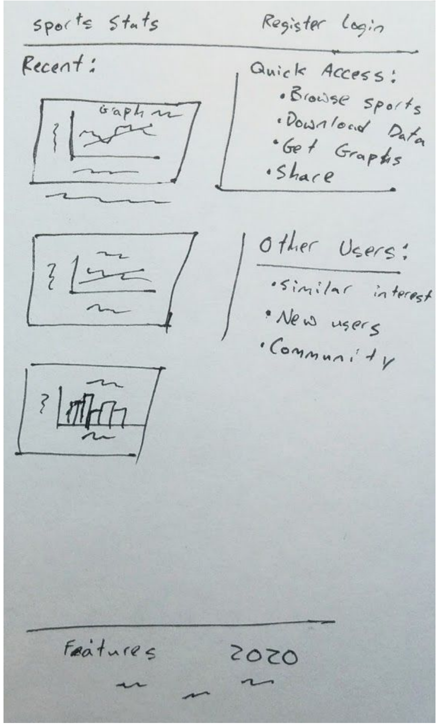
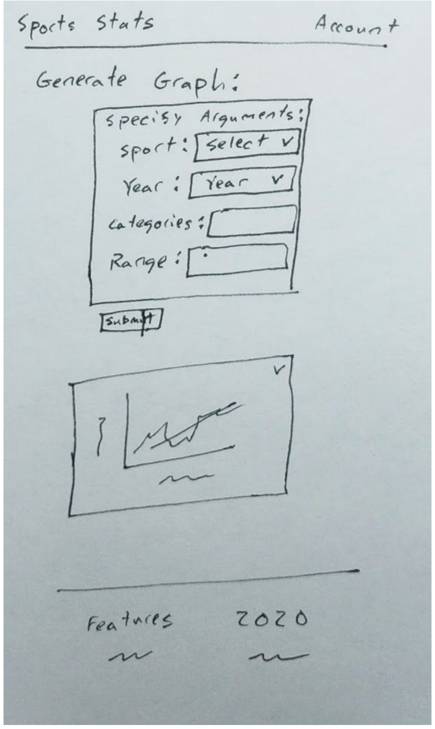
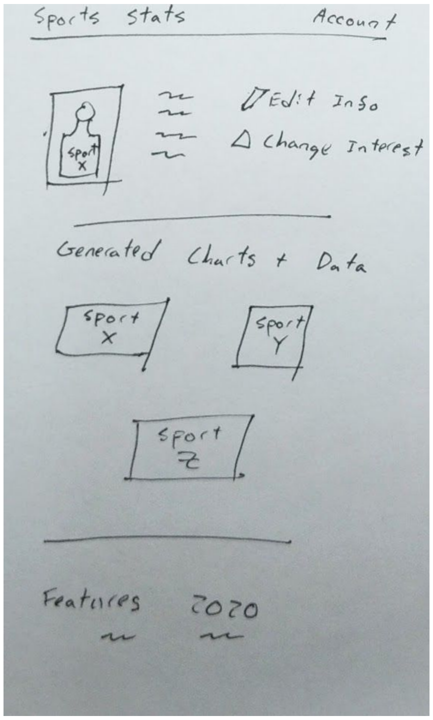
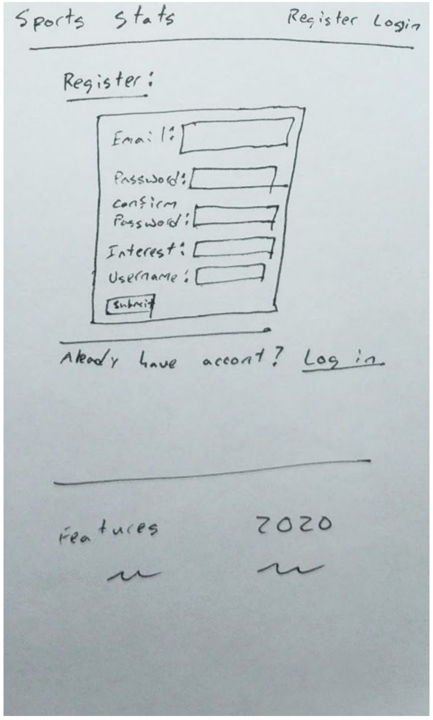
Architecture Diagram:



- The web application will be hosted through Heroku. SQLAlchemy is a database application that allows us to interface with different SQL-type relational databases in Flask easily. Through SQLAlchemy we are currently using SQLite, with the intention that we will move to postgresql in production.
- Flask acts as both the backend and integration layer where all our web logic is implemented. It allows us to use our dynamic templates and partials to generate our frontend code. The protocols used are standard in Flask framework and are handled black-box.
- We are using bcrypt to authenticate user passwords and it communicates with the database through Flask. We expect that the user will use http to communicate with the website.

Front End Design: Wireframes

- General Design Wireframes

-  
-  
- Other features such as account creation and download data functionality follow similar design principles, but are generic forms that handle data.

Web Service Design

- Our project relies on sports-reference API to get all of the data that is relayed to the client-side. Although this API is not first-party to sport-reference.com, the API is open source and well documented. The sports-reference API is called through the python package with the same name. We use this python package with imports to fetch the object that will hold the desired data. There is no form that passed, but rather the desired object (which will hold the requested data).

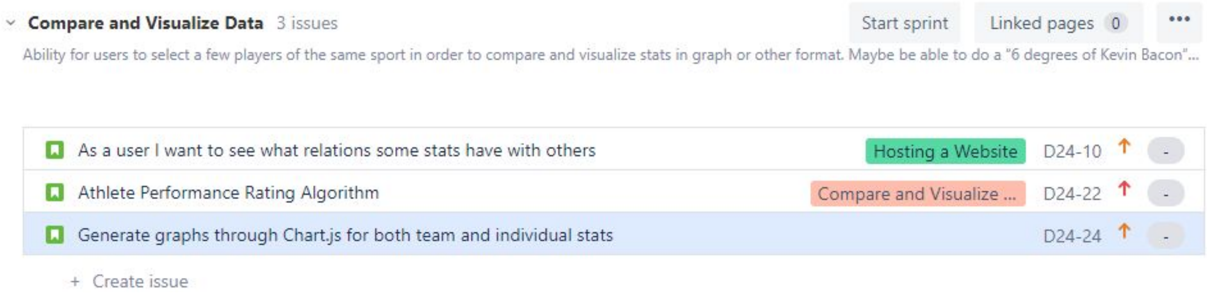
Database design

- Currently the only data being stored in the database is the user/account information as well as the sports data that the user has requested. They are tied together by the same key.
- We are using SQLAlchemy to host the database application, which is using SQL relational-database system. We plan to switch to postgreSQL in production.

Individual Contributions

- Zoe Stewart
    - Account profile creation page
- David Dayan
    - Finished profile creation functionality and download data feature. Rebuilt package structure to work with API calls and user authentication.
        - Latest Commit
- John Lee
    - Worked on registration. Regex errors, redirecting after registration/login.
- Makayla Johnson
    - Worked on the registration page
- Quinn Stone
    - Got proper flask packages installed on local machine. Added styling components to the sports pages to make for a more visually appealing UI.
        - Latest Commit
- Mitchell LaRocque
    - Worked on local flask integration in order to prepare for database integration. Worked on researching methods to utilize database with flask.

Project Management



Compare and Visualize Data 3 issues

Ability for users to select a few players of the same sport in order to compare and visualize stats in graph or other format. Maybe be able to do a "6 degrees of Kevin Bacon"...

As a user I want to see what relations some stats have with others	Hosting a Website	D24-10	-
Athlete Performance Rating Algorithm	Compare and Visualize ...	D24-22	-
Generate graphs through Chart.js for both team and individual stats		D24-24	-

-

Challenges

1. There have been stability issues with packages needed for registration. Although unlikely, if registration cannot be implemented, users should be able to use the same planned functionalities. **This has been solved.**
2. When a user requests data we will not store the entire data, but only the query or form that the user requested that procured that data. If storing these queries is not possible per user, some sort of identification for each graph should be available as an image that is generated on the backend. Not being able to do this could result in issues where the user could not access previously generated graphs.
3. Inconsistent work from team members. The risk mitigation for this is that we will have to cut more features or limit project scope.